Transform Decoding of Reed-Solomon Codes Over GF(2^{2^n}) Using the Techniques of Winograd*

I. S. Reed
University of Southern California

T. K. Truong and B. Benjauthrit
TDA Engineering Office

A new algorithm for computing a Fourier-like transform over GF(2^{2^n}), where n = 1,2,3,4,5, is developed to encode and decode Reed-Solomon (RS) codes of length 2^{2^n} Such an RS decoder is considerably faster than a decoder that uses the conventional fast transform over GF(2^{2^n}).

I. Introduction

Fast real-valued transforms over the group $(Z_2)^n$ were developed first by Green (Ref. 1) to decode the (32,6) Reed-Muller code (Ref. 2) used by JPL in the Mariner and Viking space probes. Recently Gore (Ref. 3) extended Mandelbaum's methods (Ref. 4) for decoding Reed-Solomon codes. He proposed to decode RS codes with a finite field transform over GF(2^n), where n is an integer. Michelson (Ref. 5) has implemented Mandelbaum's algorithm and showed that the decoder, using the transform over GF(2^n), requires substantially fewer multiplications than a more standard decoder (Refs. 6, 7). The disadvantage of his transform method over GF(2^n) is that the transform length is an odd number, so that the most efficient FFT algorithm cannot be used.

For a space communication link, it was shown in (Ref. 8) that the concatenated 16-error-correcting, 255-symbol RS code, each symbol with 8 bits, and a k = 7, rate = 1/2 or 1/3, Viterbi decoded convolutional code, can be used to reduce the value of E_b/N_o required to meet a specified bit-error rate P_b. Here E_b is the received energy for each bit, and N_o is the noise power spectral density at the receiver input. Such a concatenated RS – convolutional code is being considered currently by JPL for the Voyager missions.

*This work was supported in part by the U.S. Air Force Office of Scientific Research under Grant Number AFOSR 75-2798.

The Voyager RS code utilizes 255 symbols for information and error control out of a possible 256 symbols. Of the 255 RS symbols only 223 are actually used as information symbols. The remaining 32 symbols are parity check symbols. It is by this means that the 255-symbol RS code is concatenated with a k = 7, rate 1/2 or 1/3 convolutional code.

In this paper, a new algorithm based on the methods of Winograd (Refs. 9, 10) is developed to compute a transform over GF(2^8) or more generally over GF(2^{2^n}) for n = 1,2,. . . ,5. This transform algorithm over GF(2^{2^n}) for n = 1,2,3,4 requires fewer multiplications than the more conventional fast transform algorithm described by Gentleman (Ref. 11). The algorithm is presented in detail in this paper only for the cases n = 2,3. This algorithm for other values of n and for RS codes over GF(2^m) where $m \neq 2^n$ can be treated in a similar manner though perhaps not as simply.

II. A New Algorithm for Computing a Transform over GF(2^{2^n}) of $2^{2^n} - 1$ Points for n = 1, 2, . . . , 5

Let GF(2^{2^n}) be the finite field of 2^{2^n} elements. Also let N be an integer that divides $2^{2^n} - 1$. Next, let the element $\gamma \epsilon GF(2^{2^n})$ generate the cyclic subgroup of N elements, $G_N = \{\gamma, \gamma^2, \ldots, \gamma^N = 1\}$, in the multiplicative group of GF(2^{2^n}). The transform over this subgroup G_N can be defined by

$$A_j = \sum_{i=0}^{N-1} a_i \gamma^{ij}$$

for

$$0 \leq j \leq N-1$$

where

$$a_i \in GF(2^{2^n})$$

Rewrite this in matrix form as

$$\bar{A} = W'\bar{a} , \tag{1}$$

where

$$W' = (w'_{i,j})$$

and

$$w'_{i,j} = \gamma^{ij}$$

Also let

$$A_0 = \sum_{i=0}^{N-1} a_i$$

and

$$A_j = A_0 + B_j$$

for

$$j = 1,2,\ldots, N-1$$

where

$$B_j = \sum_{i=1}^{N-1} a_i \gamma^{ij}$$

That is, let

$$\overline{B} = W\overline{a} \tag{2}$$

where W is the $(N-1) \times (N-1)$ matrix $(\gamma^{ij})_{i,j \neq 0}$ and $\overline{a}, \overline{B}$ are the column matrices (a_i) and (B_k), respectively.

For $n = 1,2,\ldots,5$, the order of a multiplicative group of $GF(2^{2^n})$ can be factored into Fermat prime factors, i.e.,

$$2^{2^n} - 1 = \prod_{i=0}^{n-1} (2^{2^i} + 1)$$

If N is a Fermat prime p, one can find an element $\alpha \in GF(P)$ which generates the cyclic subgroup of p-1 elements. Hence a permutation or substitution σ can be defined by

$$\sigma = \begin{pmatrix} 1, & 2, & \ldots, & P-2, & P-1 \\ \alpha, & \alpha^2, & \ldots, & \alpha^{P-2}, & \alpha^{P-1} = 1 \end{pmatrix} \mod p$$

where all the elements of this substitution are taken modulo p.

Using the above permutation, by (Ref. 12), one can permute the indices of $\overline{B}, \overline{a}$, W defined in (2) so that the matrix $\overline{W} = (\gamma^{\sigma(i)\sigma(j)})_{i,j \neq 0}$ is cyclic. That is,

$$B_{\sigma(j)} = \sum_{i=1}^{p-1} a_{\sigma(i)} \gamma^{\sigma(i)\sigma(j)}$$

$$= \sum_{i=1}^{p-1} a_{\sigma(i)} \gamma^{\alpha^{i+j}} \tag{3a}$$

$$= \sum_{i=1}^{p-1} a_{\sigma(i)} \sigma^{\sigma(i+j)}$$

for

$$j = 1, 2, \ldots, p-1.$$

This is reexpressed in matrix form as

$$\tilde{B} = \tilde{W}\,\tilde{a} \tag{3b}$$

where

$$\tilde{B} = (B_{\sigma(j)}), \quad \tilde{W} = \left(\gamma^{\sigma(i+j)}\right)_{i,j\neq 0}$$

and

$$\tilde{a} = (a_{\sigma(i)})$$

By (3a) $B_{\sigma(j)}$ is a cyclic convolution of $a_{\sigma(i)}$ and $\gamma^{\sigma(i)}$ for $j = 1,2,\cdots,p-1$. But also (3a) is the set of coefficients of

$$T(u) = \left(\sum_{i=1}^{p-1} a_{\sigma(p-i)}\, u^{i-1}\right) \times \left(\sum_{i=1}^{p-1} \gamma^{\sigma(i)}\, u^{i-1}\right) \bmod u^{p-1} - 1$$

Since p is a Fermat prime, $u^{p-1} - 1 \equiv (u+1)^{p-1} \bmod 2$ so that one cannot factor $u^{p-1} - 1$ over GF(2) into irreducible relatively prime factors. Hence, Winograd's method (Refs. 9, 10) for using the Chinese remainder theorem to evaluate T(u) with the residues

of these factors cannot be used directly. Thus, special techniques are developed in the following sections to calculate the p-point transform over $GF(2^{2^n})$, where p is a Fermat prime.

Let

$$N = 2^{2^n} - 1 = \prod_{i=0}^{n-1} \left(2^{2^i} + 1\right) = N_1 N_2 \cdots N_k ,$$

where

$$(N_i, N_j) = 1$$

for $i \neq j$. Using the Chinese remainder theorem (Ref. 13), it is shown by Winograd in (Refs. 9, 10) that the transform matrix W' defined in (1) can be transformed into the direct product of $W'_1, W'_2, \ldots, W'_k$, where W'_i is the matrix of an N_i-point discrete Fourier-like transform. Assume the number of multiplications needed to perform an N_i-point transform over $GF(2^{2^n})$ for $i = 1,2,\ldots,k$ is m_i. Then, the number of multiplications for computing an N-point transform is $m_1 m_2 \ldots m_k$.

III. Transform Over $GF(2^4)$ of 15 Points

Consider the finite field $GF(2^4)$. Since $N = 2^4 - 1 = 3 \times 5$, the algorithm described in the previous section can be used to calculate the transform of 15 points over $GF(2^4)$. To do this the N_i-point transforms over $GF(2^{2^n})$ are first developed individually for $N_i = 3,5$. Let γ be a 3rd root of unity in $GF(2^{2^n})$ for $n = 1,2,\ldots,5$.

For $N_i = 3$, the transform over $GF(2^{2^n})$ for $n = 1,2,\ldots,5$ is expressible as

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} \gamma^0, & \gamma^0, & \gamma^0 \\ \gamma^0, & \gamma^1, & \gamma^2 \\ \gamma^0, & \gamma^2, & \gamma^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \tag{4}$$

Let

$$\begin{aligned} m_0 &= \gamma^0 (a_o + a_1 + a_2) \\ m_1 &= (a_1 + a_2) \cdot \gamma^1 \\ m_2 &= (\gamma^2 - \gamma^1) \cdot a_1 = \gamma^0 a_1 \\ m_3 &= (\gamma^2 - \gamma^1) \cdot a_2 = \gamma^0 a_2 \end{aligned} \tag{5}$$

Thus,

$$\begin{aligned} A_0 &= m_0 \\ A_1 &= m_0 + m_1 + m_2 \\ A_2 &= m_0 + m_1 + m_3 \end{aligned} \tag{6}$$

In what follows, a multiplication by the element γ^0 will need to be considered sometimes as a multiplication. Hence by (5), if one includes multiplications by the unit $\gamma^0 = 1$, the total number of multiplications needed to perform the above transform is 4.

Next consider the case $N_i = 5$. Let γ be a 5th root of unity in $GF(2^{2^n})$ for $n = 2,3, \ldots ,5$. The 5-point transform is equivalent to

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = \begin{pmatrix} \gamma^0 & \gamma^0 & \gamma^0 & \gamma^0 & \gamma^0 \\ \gamma^0 & \gamma^1 & \gamma^2 & \gamma^3 & \gamma^4 \\ \gamma^0 & \gamma^2 & \gamma^4 & \gamma^1 & \gamma^3 \\ \gamma^0 & \gamma^3 & \gamma^1 & \gamma^4 & \gamma^2 \\ \gamma^o & \gamma^4 & \gamma^3 & \gamma^2 & \gamma^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} \tag{7}$$

Thus,

$$A_0 = \gamma^0 (a_o + a_1 + a_2 + a_3 + a_4 + a_5)$$

and

$$\overline{B} = (B_i) = \begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{pmatrix} = \begin{pmatrix} \gamma^1 & \gamma^2 & \gamma^3 & \gamma^4 \\ \gamma^2 & \gamma^4 & \gamma^1 & \gamma^3 \\ \gamma^3 & \gamma^1 & \gamma^4 & \gamma^2 \\ \gamma^4 & \gamma^3 & \gamma^2 & \gamma^1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}$$

$$= W\overline{a} = (w_{ij}) (a_i)$$

Let

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2^2 & 2^3 & 2^4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} \mod 5$$

Now

$$(B_{\sigma(i)}) = (w_{\sigma(i),\,\sigma(j)})\, a_{\sigma(i)}$$

or

$$\tilde{B} = \begin{pmatrix} B_2 \\ B_4 \\ B_3 \\ B_1 \end{pmatrix} = \begin{pmatrix} w_{2,2} & w_{2,4} & w_{2,3} & w_{2,1} \\ w_{4,2} & w_{4,4} & w_{4,3} & w_{4,1} \\ w_{3,2} & w_{3,4} & w_{3,3} & w_{3,1} \\ w_{1,2} & w_{1,4} & w_{1,3} & w_{1,1} \end{pmatrix} \begin{pmatrix} a_2 \\ a_4 \\ a_3 \\ a_1 \end{pmatrix} \quad (8)$$

$$= \begin{pmatrix} \gamma^4 & \gamma^3 & \gamma^1 & \gamma^2 \\ \gamma^3 & \gamma^1 & \gamma^2 & \gamma^4 \\ \gamma^1 & \gamma^2 & \gamma^4 & \gamma^3 \\ \gamma^2 & \gamma^4 & \gamma^3 & \gamma^1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_4 \\ a_3 \\ a_1 \end{pmatrix} = \tilde{W}\tilde{a}$$

This is of form

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} A & B \\ B & A \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

where

$$Y_1 = \begin{pmatrix} B_2 \\ B_4 \end{pmatrix}, \quad Y_2 = \begin{pmatrix} B_3 \\ B_1 \end{pmatrix},$$

$$X_1 = \begin{pmatrix} a_2 \\ a_4 \end{pmatrix}, \quad X_2 = \begin{pmatrix} a_3 \\ a_2 \end{pmatrix},$$

$$A = \begin{pmatrix} \gamma^4 & \gamma^3 \\ \gamma^3 & \gamma^1 \end{pmatrix}, \quad B = \begin{pmatrix} \gamma^1 & \gamma^2 \\ \gamma^2 & \gamma^4 \end{pmatrix}$$

Thus

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} A\,X_1 + B\,X_2 \\ B\,X_1 + A\,X_2 \end{pmatrix} = \begin{pmatrix} A\,(X_1 + X_2) + (B - A)\,X_2 \\ A\,(X_1 + X_2) + (B - A)\,X_1 \end{pmatrix}$$

$$= \begin{pmatrix} D + E \\ D + F \end{pmatrix}$$

where

$$D = A\,(X_1 + X_2), \quad E = (B - A)\,X_2,$$

$$F = (B - A)\,X_1$$

Now

$$D = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = A\,(X_1 + X_2)$$

$$= \begin{pmatrix} \gamma^4 & \gamma^3 \\ \gamma^3 & \gamma^1 \end{pmatrix} \begin{pmatrix} a_2 + a_3 \\ a_4 + a_1 \end{pmatrix}$$

$$= \begin{pmatrix} \gamma^3\,(a_1 + a_2 + a_3 + a_4) + (\gamma^4 + \gamma^3)\,(a_2 + a_3) \\ \gamma^3\,(a_1 + a_2 + a_3 + a_4) + (\gamma^1 + \gamma^3)\,(a_4 + a_1) \end{pmatrix}$$

$$E = (B - A)(X_2) = \begin{pmatrix} \gamma^4 - \gamma^1, & \gamma^3 - \gamma^2 \\ \gamma^3 - \gamma^2, & \gamma^1 - \gamma^4 \end{pmatrix} \begin{pmatrix} a_3 \\ a_1 \end{pmatrix}$$

$$= \begin{pmatrix} (\gamma^1 + \gamma^4)(a_1 + a_3) + \gamma^0 a_1 \\ (\gamma^1 + \gamma^4)(a_2 + a_4) + \gamma^0 a_3 \end{pmatrix}$$

Similarly

$$F = \begin{pmatrix} (\gamma^1 + \gamma^4)(a_2 + a_4) + \gamma^0 a_4 \\ (\gamma^1 + \gamma^4)(a_2 + a_4) + \gamma^0 a_2 \end{pmatrix}$$

Thus

$$\begin{pmatrix} B_2 \\ B_4 \\ B_3 \\ B_1 \end{pmatrix} = \begin{pmatrix} \gamma^3 (a_1 + a_2 + a_3 + a_4) + (\gamma^3 + \gamma^4) \cdot (a_2 + a_3) + (\gamma^1 + \gamma^4)(a_1 + a_3) + \gamma^0 a_1 \\ \gamma^3 (a_1 + a_2 + a_3 + a_4) + (\gamma^1 + \gamma^3) \cdot (a_1 + a_4) + (\gamma^1 + \gamma^4)(a_1 + a_3) + \gamma^0 a_3 \\ \gamma^3 (a_1 + a_2 + a_3 + a_4) + (\gamma^3 + \gamma^4)(a_2 + a_3) + (\gamma^1 + \gamma^4)(a_2 + a_4) + \gamma^0 a_4 \\ \gamma^3 (a_1 + a_2 + a_3 + a_4) + (\gamma^1 + \gamma^3)(a_1 + a_4) + (\gamma + \gamma^4)(a_2 + a_4) + \gamma^0 a_2 \end{pmatrix}$$

Hence

$$A_1 = \gamma^0 a_0 + B_1 = \gamma^0 a_0 + \gamma^3 (a_1 + a_2 + a_3 + a_4) + (\gamma^1 + \gamma^3)(a_1 + a_4) + (\gamma + \gamma^4)(a_2 + a_4) + \gamma^0 a_2$$

$$= \gamma^0 (a_0 + a_1 + a_2 + a_3 + a_4) + (\gamma^0 + \gamma^3)(a_1 + a_2 + a_3 + a_4) + (\gamma^1 + \gamma^3)(a_1 + a_4) + (\gamma + \gamma^4)(a_2 + a_4) + \gamma^0 a_2$$

$$A_2 = \gamma^0 (a_0 + a_1 + a_2 + a_3 + a_4) + (\gamma^0 + \gamma^3)(a_2 + a_3 + a_4 + a_1) + (\gamma^3 + \gamma^4)(a_2 + a_3) + (\gamma + \gamma^4)(a_3 + a_1) + \gamma^0 a_1$$

$$A_3 = \gamma^0 (a_0 + a_1 + a_2 + a_3 + a_4) + (\gamma^0 + \gamma^3) (a_2 + a_3 + a_4 + a_1) + (\gamma^3 + \gamma^4) (a_2 + a_3) + (\gamma + \gamma^4) (a_2 + a_4) + \gamma^0 a_4$$

$$A_4 = \gamma^0 (a_0 + a_1 + a_2 + a_3 + a_4) + (\gamma^0 + \gamma^3) (a_2 + a_3 + a_4 + a_1) + (\gamma^1 + \gamma^3) (a_1 + a_4) + (\gamma + \gamma^4) (a_1 + a_3) + \gamma^0 a_3$$

Now let

$$\begin{aligned}
m_0 &= \gamma^0 (a_0 + a_1 + a_2 + a_3 + a_4) \\
m_1 &= (\gamma^0 + \gamma^3) \cdot (a_1 + a_2 + a_3 + a_4) \\
m_2 &= (\gamma^3 + \gamma^4) \cdot (a_2 + a_3) \\
m_3 &= (\gamma^1 + \gamma^3) \cdot (a_1 + a_4) \\
m_4 &= (\gamma + \gamma^4) \cdot (a_1 + a_3) \\
m_5 &= (\gamma + \gamma^4) \cdot (a_2 + a_4) \\
m_6 &= \gamma^0 \cdot a_1 \\
m_7 &= \gamma^0 \cdot a_3 \\
m_8 &= \gamma^0 \cdot a_4 \\
m_9 &= \gamma^0 \cdot a_2 \\
S_1 &= m_0 + m_1 + m_2 \\
S_2 &= m_0 + m_1 + m_3
\end{aligned} \tag{9}$$

Then

$$\begin{aligned}
A_0 &= m_0 \\
A_1 &= S_2 + m_5 + m_9 \\
A_2 &= S_1 + m_4 + m_6 \\
A_3 &= S_1 + m_5 + m_8 \\
A_4 &= S_2 + m_4 + m_7
\end{aligned} \tag{10}$$

If again one includes multiplications by the unit γ^0, it follows from the algorithm in (9) that the number of integer multiplications needed to perform a 5-point transform is 10. If multiplications by γ^0 are excluded, evidently only 5 multiplies are actually needed.

Now consider the case $N = 15 = N_1 N_2 = 3 \cdot 5$. Let integer $0 \leqslant i < 15$ be represented by a pair $(i_1, i_2) = (i \bmod 3, i \bmod 5)$. Since $(3,5) = 1$, by the Chinese remainder theorem,

$$i \equiv (i_1 \cdot 10 + i_2 \cdot 6) \bmod 15 \tag{11}$$

Let γ_1 and γ_2 be 3rd and 5th roots of unity in $GF(2^4)$, respectively. The 15-point transform over $GF(2^4)$ in i_1 and i_2 is

$$A_j = \sum_{i=0}^{15} a_i \gamma^{ij}$$

or

$$\begin{aligned} A_{(j_1,j_2)} &= \sum_{i_1=0}^{3-1} \left[\sum_{i_2=0}^{5-1} a_{(i_1,i_2)} \gamma_2^{i_2 \cdot j_2} \right] \gamma_1^{i_1 \cdot j_1} \\ &= \sum_{i_1=0}^{2} a_{i_1}(j_2) \gamma_1^{i_1 \cdot j_1} \end{aligned} \tag{12}$$

where

$$a_{i_1}(j_2) = \sum_{i_2=0}^{5-1} a_{(i_1,i_2)} \gamma_2^{i_2 \cdot j_2},$$

$$j_1 = 0,1,2, \quad \text{and}$$

$$j_2 = 0,1,\ldots, 4$$

or in matrix notation,

$$\left(a_{i_1}(j_2)\right) = W_2' \bar{a}_{i_1}$$

where

$$W_2' = \begin{pmatrix} \gamma_2^0 & \gamma_2^0 & \gamma_2^0 & \gamma_2^0 & \gamma_2^0 \\ \gamma_2^0 & \gamma_2^1 & \gamma_2^2 & \gamma_2^3 & \gamma_2^4 \\ \gamma_2^0 & \gamma_2^2 & \gamma_2^4 & \gamma_2^1 & \gamma_2^3 \\ \gamma_2^0 & \gamma_2^3 & \gamma_2^1 & \gamma_2^4 & \gamma_2^2 \\ \gamma_2^0 & \gamma_2^4 & \gamma_2^3 & \gamma_2^2 & \gamma_2^1 \end{pmatrix}, \quad \bar{a}_{i_1} = \begin{pmatrix} a_{(i_1,0)} \\ a_{(i_1,1)} \\ a_{(i_1,2)} \\ a_{(i_1,3)} \\ a_{(i_1,4)} \end{pmatrix}$$

Thus (12) becomes

$$\bar{A}_{j_1} = \sum_{i_1=0}^{2} \gamma_1^{i_1 j_1} W_2' \bar{a}_{i_1} \tag{13}$$

for

$$j_1 = 0,1,2$$

or

$$\begin{pmatrix} \bar{A}_0 \\ \bar{A}_1 \\ \bar{A}_2 \end{pmatrix} = \begin{pmatrix} W_2' & W_2' & W_2' \\ W_2' & W_2'\gamma_1 & W_2'\gamma_1^2 \\ W_2' & W_2'\gamma_1^2 & W_2'\gamma_1 \end{pmatrix} \begin{pmatrix} \bar{a}_0 \\ \bar{a}_1 \\ \bar{a}_2 \end{pmatrix}$$

Now by (11), one obtains $\bar{A}_0$ in terms of A_k:

$$\bar{A}_0 = \begin{pmatrix} A_{(0,0)} \\ A_{(0,1)} \\ A_{(0,2)} \\ A_{(0,3)} \\ A_{(0,4)} \end{pmatrix} = \begin{pmatrix} A_0 \\ A_6 \\ A_{12} \\ A_3 \\ A_9 \end{pmatrix}$$

Similarly

$$\bar{A}_1 = \begin{pmatrix} A_{10} \\ A_1 \\ A_7 \\ A_{13} \\ A_4 \end{pmatrix}, \quad \bar{A}_2 = \begin{pmatrix} A_5 \\ A_{11} \\ A_2 \\ A_8 \\ A_{14} \end{pmatrix}$$

and

$$\bar{a}_0 = \begin{pmatrix} a_0 \\ a_6 \\ a_{12} \\ a_3 \\ a_9 \end{pmatrix}, \quad \bar{a}_1 = \begin{pmatrix} a_{10} \\ a_1 \\ a_7 \\ a_{13} \\ a_4 \end{pmatrix}, \quad \bar{a}_2 = \begin{pmatrix} a_5 \\ a_{11} \\ a_2 \\ a_8 \\ a_{14} \end{pmatrix}$$

Using the 3-point transform in (4) and making the correspondances, $\gamma^0 \leftrightarrow W_2'$, $\gamma^1 \leftrightarrow W_2' \gamma_1$, $\gamma^2 \leftrightarrow W_2' \gamma_1^2$, one obtains

$$M_0 = W_2' \cdot (\bar{a}_0 + \bar{a}_1 + \bar{a}_2) = W_2' \begin{pmatrix} a_0 + a_{10} + a_5 \\ a_6 + a_1 + a_{11} \\ a_{12} + a_7 + a_2 \\ a_3 + a_{13} + a_8 \\ a_9 + a_4 + a_{14} \end{pmatrix}, \tag{14}$$

$$M_1 = W_2' \gamma_1 (\bar{a}_1 + \bar{a}_2) = W_2' \begin{pmatrix} \gamma_1 (a_{10} + a_5) \\ \gamma_1 (a_1 + a_{11}) \\ \gamma_1 (a_7 + a_2) \\ \gamma_1 (a_{13} + a_8) \\ \gamma_1 (a_4 + a_{14}) \end{pmatrix},$$

$$M_2 = W_2' \cdot \bar{a}_1 \ ,$$

$$M_3 = W_2' \cdot \bar{a}_2$$

Equation (14) requires 4 matrix multiplies. Thus,

$$\begin{aligned} \bar{A}_0 &= M_0 \ , \\ \bar{A}_1 &= M_0 + M_1 + M_2 \ , \end{aligned} \tag{15}$$

and

$$\bar{A}_3 = M_0 + M_1 + M_3$$

Observe that all four matrix multiplies in (14) are 5-point transforms of exactly the same form as (7). Thus one may compute M_j for j = 0,1,2,3 in (14) with a procedure similar to that used to compute the matrix defined in (7).

The number of multiplications for computing an M_j for j = 0,1,2,3 in (14) is 5 excluding multiplication by γ^0. Thus, the total number of multiplications needed is $4 \times 5 = 20$.

IV. Transform Over GF(2^8) of 255 Points

Since $N = 255 = 3 \cdot 5 \cdot 17 = N_1 \cdot N_2 \cdot N_3$, by Winograd's algorithm, one needs to compute an N_i-point transform over GF(2^8) for $N_i = 3,5,17$. An N_i-transform over GF(2^8) for $N_i = 3$ or 5 was computed in the last section. For $N_i = 17$, the permutation σ is given by

$$\sigma = \begin{pmatrix} 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & 9, & 10, & 11, & 12, & 13, & 14, & 15, & 16 \\ 5, & 8, & 6, & 13, & 14, & 2, & 10, & 16, & 12, & 9, & 11, & 4, & 3, & 15, & 7, & 1 \end{pmatrix}$$

Applying the above permutation to (2), one obtains a 16×16 cyclic matrix. By theorem 1 in Appendix A, the cyclic matrix can be partitioned into blocks of 4×4 matrices so that the blocks form a 4×4 cyclic matrix. This has the form

$$\begin{pmatrix} T_2 \\ T_4 \\ T_3 \\ T_1 \end{pmatrix} = \begin{pmatrix} A & B & C & D \\ B & C & D & A \\ C & D & A & B \\ D & A & B & C \end{pmatrix} \begin{pmatrix} S_2 \\ S_4 \\ S_3 \\ S_1 \end{pmatrix} \tag{16}$$

where

$$T_2 = \begin{pmatrix} b_5 \\ b_8 \\ b_6 \\ b_{13} \end{pmatrix}, \qquad T_4 = \begin{pmatrix} b_{14} \\ b_2 \\ b_{10} \\ b_{16} \end{pmatrix},$$

$$T_3 = \begin{pmatrix} b_{12} \\ b_9 \\ b_{11} \\ b_4 \end{pmatrix}, \qquad T_1 = \begin{pmatrix} b_3 \\ b_5 \\ b_7 \\ b_1 \end{pmatrix},$$

$$A = \begin{pmatrix} \gamma^8 & \gamma^6 & \gamma^{13} & \gamma^{14} \\ \gamma^6 & \gamma^{13} & \gamma^{14} & \gamma^2 \\ \gamma^{13} & \gamma^{14} & \gamma^2 & \gamma^{10} \\ \gamma^{14} & \gamma^2 & \gamma^{10} & \gamma^{16} \end{pmatrix},$$

$$B = \begin{pmatrix} \gamma^2 & \gamma^{10} & \gamma^{16} & \gamma^{12} \\ \gamma^{10} & \gamma^{16} & \gamma^{12} & \gamma^9 \\ \gamma^{16} & \gamma^{12} & \gamma^9 & \gamma^{11} \\ \gamma^{12} & \gamma^9 & \gamma^{11} & \gamma^4 \end{pmatrix},$$

$$C = \begin{pmatrix} \gamma^9 & \gamma^{11} & \gamma^4 & \gamma^3 \\ \gamma^{11} & \gamma^4 & \gamma^3 & \gamma^{15} \\ \gamma^4 & \gamma^3 & \gamma^{15} & \gamma^7 \\ \gamma^3 & \gamma^{15} & \gamma^7 & \gamma^1 \end{pmatrix},$$

$$D = \begin{pmatrix} \gamma^{15} & \gamma^{7} & \gamma^{1} & \gamma^{5} \\ \gamma^{7} & \gamma^{1} & \gamma^{5} & \gamma^{8} \\ \gamma^{1} & \gamma^{5} & \gamma^{8} & \gamma^{6} \\ \gamma^{5} & \gamma^{8} & \gamma^{6} & \gamma^{13} \end{pmatrix},$$

$$S_2 = \begin{pmatrix} a_5 \\ a_8 \\ a_6 \\ a_{13} \end{pmatrix}, \qquad S_4 = \begin{pmatrix} a_{14} \\ a_2 \\ a_{10} \\ a_{16} \end{pmatrix},$$

$$S_3 = \begin{pmatrix} a_{12} \\ a_9 \\ a_{11} \\ a_4 \end{pmatrix}, \qquad S_1 = \begin{pmatrix} a_3 \\ a_{15} \\ a_7 \\ a_1 \end{pmatrix}.$$

Now if one makes the correspondences, $A \leftrightarrow \gamma^4$, $B \leftrightarrow \gamma^3$, $C \leftrightarrow \gamma$, $D \leftrightarrow \gamma^2$, and $I_0 = A + B + C + D \leftrightarrow \gamma + \gamma^2 + \gamma^3 + \gamma^4 = 1$ in (8), then by a procedure similar to that used to compute the matrix defined in (8) one obtains

$$\begin{aligned} N_1 &= B \cdot (S_1 + S_2 + S_3 + S_4) \\ N_2 &= (A + B) \cdot (S_2 + S_3) \\ N_3 &= (C + B) \cdot (S_1 + S_4) \\ N_4 &= (C + A) \cdot (S_3 + S_1) \\ N_5 &= (C + A) \cdot (S_2 + S_4) \\ N_6 &= I_0 \cdot S_1 \\ N_7 &= I_0 \cdot S_3 \\ N_8 &= I_0 \cdot S_4 \\ N_9 &= I_0 \cdot S_2 \end{aligned} \tag{17}$$

and

$$V_1 = N_1 + N_2$$

$$V_2 = N_1 + N_3$$

Equation (17) requires 9 (4 X 4) matrix multiplies. Then

$$\begin{aligned} T_2 &= V_1 + N_4 + N_6 \\ T_4 &= V_2 + N_4 + N_7 \\ T_3 &= V_1 + N_5 + N_8 \\ T_1 &= V_2 + N_5 + N_9 \end{aligned} \tag{18}$$

To find $N_1, \ldots N_9$, one needs to multiply matrices of form (A + B), (C + B), etc., by vectors $(S_i + S_j)$, S_i, etc. For example, consider $N_2 = (A + B) \cdot (S_2 + S_3)$,

$$N_2 = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} \gamma^8+\gamma^2, & \gamma^6+\gamma^{10}, & \gamma^{13}+\gamma^{16}, & \gamma^{14}+\gamma^{12} \\ \gamma^6+\gamma^{10}, & \gamma^{13}+\gamma^{16}, & \gamma^{14}+\gamma^{12}, & \gamma^2+\gamma^9 \\ \gamma^{13}+\gamma^{16}, & \gamma^{14}+\gamma^{12}, & \gamma^2+\gamma^9, & \gamma^{10}+\gamma^{11} \\ \gamma^{14}+\gamma^{12}, & \gamma^2+\gamma^9, & \gamma^{10}+\gamma^{11}, & \gamma^{16}+\gamma^4 \end{pmatrix} \begin{pmatrix} a_5 + a_{12} \\ a_8 + a_9 \\ a_6 + a_{11} \\ a_{13} + a_4 \end{pmatrix} \tag{19}$$

$$= \begin{pmatrix} J, & K \\ K, & L \end{pmatrix} \begin{pmatrix} E_0 \\ E_1 \end{pmatrix}$$

where J, K, L are 2 X 2 matrices. Hence

$$N_2 = \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} = \begin{pmatrix} J & K \\ K & L \end{pmatrix} \begin{pmatrix} E_0 \\ E_1 \end{pmatrix} = \begin{pmatrix} (E_0+E_1)K + (J+K)E_0 \\ (E_0+E_1)K + (L+K)E_0 \end{pmatrix}$$

Let

$$\begin{aligned} U_1 &= (E_0+E_1) \cdot K \\ U_2 &= (J+K) \cdot E_0 \\ U_3 &= (L+K) \cdot E_0 \end{aligned} \tag{20}$$

Then $F_0 = U_1 + U_2$, $F_4 = U_1 + U_3$. Note that 3 (2 X 2) matrix multiplies are necessary to perform (20). The matrix U_1 in (20) is given by the relationship,

$$U_1 = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \gamma^{13}+\gamma^{16}, \gamma^{14}+\gamma^{12} \\ \gamma^{14}+\gamma^{12}, \gamma^{2}+\gamma^{9} \end{pmatrix} \begin{pmatrix} a_5+a_{12}+a_6+a_{11} \\ a_8+a_9+a_{13}+a_4 \end{pmatrix} \tag{21}$$

$$= \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} b \cdot (k_1+k_2) + (a+b) \cdot k_1 \\ b \cdot (k_1+k_2) + (b+c)\, k_1 \end{pmatrix}$$

Equation (21) requires 3 multiplies. In a similar manner, the matrices U_2 and U_3 in (20) can be obtained, using 3 multiplications. Thus the total number of multiplications needed to perform (19) is 9.

In a similar fashion, matrices N_1 and N_i for $i = 3, \ldots, 9$ given in (17) can also be obtained, each requiring 9 multiplications. Hence, the number of multiplications needed to perform a 17-point transform over $GF(2^{2^n})$ for $n = 3,4,5$ is $9 \times 5 = 45$, excluding multiplications by γ^0. To include multiplications by γ^0, it follows from (18) that the total number of multiplications needed is $9 \times 9 + 1 = 82$.

By the same procedure used to compute the 15-point transform over $GF(2^4)$ in the last section, the total number of multiplications needed to perform a 255-point transform over $GF(2^8)$ is $4 \times 10 \times 45 = 1800$ multiplies. If $N = 2^{2^n} - 1 = N_1 \cdot N_2 \cdots N_k$, where N_i is a prime number for $i = 1,2, \ldots, k$, then the total number of multiplications needed to perform a $(2^{2^n} - 1)$ - point transform is

$$M = \prod_{i=0}^{k-2} (3^{2^i} + 1) + 5 \times 3^{2^{n-1}-2}$$

By the principle of mathematical induction one can show that

$$\prod_{i=0}^{n} m^{2^i} + 1 = \frac{m^{2^{n+1}} - 1}{m-1}$$

for any integers $m > 1$ and $n \geq 0$. Thus

$$M = (3^{2^{k-1}} - 1)/2 + 5 \times 3^{2^{n-1}} - 2$$

By using a more conventional fast transform technique, Gentleman shows (Refs. 5, 11) that an N-point transform of such an N requires $N(N_1 + N_2 + \cdots + N_k - k + 1)$ multiplications, including multiplications by unity. The present algorithm for computing the $(2^{2^n} - 1)$ - point transform for $n = 1,2, \ldots, 5$ and Gentleman's algorithm are compared in Table 1. The number of multiplications needed to perform these algorithms

is given for both cases. From Table 1, one sees that for n = 1,2,3,4, the new algorithm for computing the $(2^{2^n} - 1)$ - point transform requires fewer multiplications than Gentleman's algorithm. This is not true for n = 5, however.

V. Transform Decoder for Reed-Solomon Codes

It is shown in (Ref. 14) that RS codes can be decoded with a fast transform algorithm over GF(p^n) and continued fractions. There it was shown that the decoding of RS codes with a finite field transform over GF(p^n), where p is a prime and n is an integer, was composed of the following 4 steps:

(a) Compute the transform over GF(p^n) of the received N-tuple,

$$E_k = \sum_{i=0}^{N-1} \gamma_i (\alpha^i)^k = \sum_{i=1}^{t} Y_i X_i^k$$

for k = 1,2, . . . 2t

where t is the number of errors, Y_i is the i-th error amplitude and $X_i = \alpha^i$ is the i-th error position.

(b) Define the generating function of the sequence (E_k) as a formal power series. That is,

$$E(x) = E_1 x^{-1} + E_2 x^{-2} + \cdots + E_{2t} x^{-2t} + \cdots$$

$$= \frac{\sum_{i=0}^{t-1} P_i x^i}{x^t + \sum_{k=1}^{t} (-1)^k \sigma_k x^{t-k}} = \frac{P(x)}{\sigma(x)}$$

Use Berlekamp's algorithm implemented by continued fraction approximations to determine the error locator polynomial $\sigma(x)$ and error evaluator polynomial P(x) from the known E_j for j = 1,2, . . . ,2t.

(c) Use these polynomials to compute the remaining transform of the error vector e_0, $e_1, \ldots, e_{N-1}$.

(d) Invert the transform to recover the error vector and then obtain the corrected code.

In order to compare the above transform decoder with the standard RS decoder, the steps used in the standard RS decoder are stated as follows:

(1) Syndrome calculation

(2) Berlekamp's algorithm or continued fractions to determine $\sigma(x)$

(3) Chien's algorithm for finding the error locations

(4) Compute error magnitudes.

One observes that step 3 in both approaches are equivalent. It is shown in (Ref. 5) that step 3 in the transform decoder requires approximately the same number of multiplications as step 3 in the standard decoder. Step 1 and step 4 in the standard decoder require more multiplications than those in the transform decoder over $GF(2^{2^n})$. To see this, let N be the block length of the RS code in $GF(2^{2^n})$. Also let $d = 2t + 1$ be the minimum distance of the code, where t is the number of allowable errors. It follows from (Refs. 5 and 15) that the number of multiplications required to perform the syndrome and error magnitude calculations for the standard decoder is approximately $(N - 1)(d - 1) + t^2$. (Note that the performance of the conventional decoder is dependent on the number of allowable errors.)

For a $(2^{2^n} - 1)$ - symbol, 2^n - symbol error - correcting, RS code for $n = 2,3$, the number of multiplications needed to compute the syndrome and the error magnitudes is given in Table 2. The new algorithm, Gentleman's algorithm, and the standard algorithm are compared in Table 2 in terms of the number of multiplications needed to compute the syndrome and the error magnitudes for decoding these RS codes.

Acknowledgment

The authors wish to thank Dr. N. A. Renzetti, Manager of Tracking and Data Acquisition Engineering, and the members of the Advanced Engineering Group in that organization at the Jet Propulsion Laboratory for their early support, suggestions, and encouragement of the research which led to this paper. We also thank Dr. C. A. Greenhall for his mathematical suggestions.

References

1. Green, R. R., "Analysis of a Serial Orthogonal Decoder," *Space Programs Summary 37-53*, Vol. III, 1968, pp. 185-187.

2. Reed, I. S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IRE Trans.*, PGIT-4, 1954, pp. 38-49.

3. Gore, W. C., "Transmitting Binary Symbols with Reed-Solomon Code," Johns Hopkins EE Report No. 73-5, April 1973.

4. Mandelbaum, D., "On Decoding Reed-Solomon Codes," *IEEE Trans. on Inform. Theory*, Vol. IT-17, No. 6, pp. 707-712, November 1971.

5. Michelson, A., "A New Decoder for the Reed-Solomon Codes Using a Fast Transform Technique," Systems Engineering Technical Memorandum, No. 52, Electronic Systems Group Eastern Division GTE Sylvania, August 1975.

6. Peterson, W. W., "Error-Correcting Codes," MIT Press, Cambridge, Mass., 1961, pp. 168-169.

7. Lin, S., *An Introduction to Error-Correcting Codes,* Englewood Cliffs, N.J., Prentice-Hall, 1970.

8. Odenwalder, J., et al., "Hybrid Coding Systems Study Final Report," Linkabit Corp., NASA CR 114,486, Sept. 1972.

9. Winograd, S., "On Computing the Discrete Fourier Transform," *Proc. Nat. Acad. Sci. USA,* 1976, 73, pp. 1005-1006.

10. Winograd, S., "On Computing the Discrete Fourier Transform," Research Report, Math. Science Dept., IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 10592.

11. Gentleman, W. M., "Matrix Multiplication and Fast Fourier Transforms," *Bell System Technical Journal,* 1968, pp. 1099-1103.

12. Reed, S. I., and Truong, T. K., "Fast Mersenne Prime Transforms for Digital Filters," to be published in the Proceedings IEE.

13. Niven, I., and Zuckerman, H. S., *An Introduction to the Theory of Numbers,* John Wiley and Sons, Inc., New York, 1972, p. 49.

14. Reed, I. S., Scholtz, R. A., Truong, T. K., and Welch, L. R., "The Fast Decoding of Reed-Solomon Codes Using Fermat Theoretic Transforms and Continued Fractions," to be published in *IEEE Trans. on Inform. Theory.*

15. Forney, G. D., "On Decoding BCH Codes," *IEEE Transactions on Information Theory,* IT-11, October 1965.

Table 1. The complexity of transform over GF(2^{2^n}) for n = 1, 2, . . . , 5

$N = 2^{2^n} - 1$	Factors $N_1 \cdot N_2 \cdots N_k = \prod_{i=0}^{k} (2^2 + 1)$	No. mult. of new algorithm $(3^{2^{k-1}} - 1)/2 + 5 \times 3^{2^{n-1}-2}$ for $n > 1$	No. mult. of Gentleman's algorithm $N(N_1 + N_2 + \ldots + N_k - k + 1)$
$2^2 - 1$	3	1	9
$2^{2^2} - 1$	3 × 5	4 × 5 = 20	15(3 + 5 − 1) = 105
$2^{2^3} - 1$	3 × 5 × 17	4 × 10 × 45 = 1800	255(3 + 5 + 17 − 2) = 5865
$2^{2^4} - 1$	3 × 5 × 17 × 257	4 × 10 × 82 × 3645 = 11,955,600	65535(3 + 5 + 17 + 257 − 3) = 18,284,265
$2^{2^5} - 1$	3 × 5 × 17 × 257 × 65537	4 × 10 × 82 × 6562 × 23914845 = 514,727,818,279,200	$(2^{32} - 1)$(3 + 5 + 17 + 257 + 65537 − 4) = 282,673,272, 520,425

Table 2. The complexity of decoding RS of $2^{2^n} - 1$ points for n = 2, 3

N	Factors $N_1, N_2 \ldots N_k$	No. mult. of new algorithm	No. mult. of Gentleman's algorithm $2N(N_1 + N_2 + \cdots + N_k - k + 1)$	No. mult. of the standard algorithm $(N - 1)(d - 1) + t^2$
15	3 × 5	2 × 20 = 40	2 × 105 = 210	$(14)(8) + 4^2 = 128$
255	3 × 5 × 17	2 × 1800 = 3600	2 × 5862 = 11724	$(254)(32) + 16^2 = 8384$

Appendix A

Definition: A matrix A(i,j),i,j ϵ Z_n is cyclic if for some function f on Z_n,

$$A(i,j) = f.((i+j) \bmod n)$$

where

$$Z_n = \{1,2,..,n\}$$

Theorem 1: Let n = ab. If A(i,j) is a cyclic matrix, then A(i,j) is a matrix of b X b submatrices such that the submatrices form an a X a cyclic matrix.

Proof: Let $A_{ij}(k,\ell)$ be (k,ℓ) - th element of the (i,j) - th b X b submatrix of **A**, where $i,j \epsilon Z_a$, $k,\ell \epsilon Z_b$.

Then

$$A_{ij}(k,\ell) = A(bi+k, bj+\ell)$$

$$= f((b(i+j)+k+\ell) \bmod ab)$$

For $i \epsilon Z_a$, define the matrix $(G_i(k,\ell))$ by

$$G_i(k,\ell) = f((bi+k+\ell) \bmod ab)$$

for $k,\ell \epsilon Z_b$. Since $b[(i+j) \bmod a] \equiv b(i+j) \bmod ab$, we have

$$A_{ij}(k,\ell) = G_{(i+j) \bmod a}(k,\ell)$$

Therefore, the a X a array of submatrices A_{ij} is cyclic.